REMARKS

Applicant thanks the Examiner for extending the courtesy of an interview on December 4, 2003. As requested, the content of the interview is summarized below.

The Rejection Under 35 U.S.C. § 112

Claim 1 as amended now reflects proper antecedent basis.

The Rejections Under 35 U.S.C. § 103(a)

As stated in the aforementioned interview, Applicant respectfully submits that the cited art does not disclose all elements of the claims as presented. More specifically, *Nihart*, *Newkerk*, and *Ismael* do not disclose creating a protocol-specific object, or returning this object to a CIM object manager.

Nihart

Nihart teaches a system for managing a computer operating system. The system has a Protocol Engine 122 and a Common Agent interface 130 (Col. 5:31-66). However, the system of *Nihart* is only configured to translate instructions such as procedure calls. More specifically, procedure calls 132 are sent to the Protocol Engine 122, where they are translated into the language used by the Common Agent interface 130 (Id.). The Common Agent interface 130 then passes these commands to the appropriate Target Component 112 for execution (Id.).

The system of *Nihart* thus translates procedure calls. *Nihart* does not disclose the creating of a protocol-specific object. Accordingly, it also cannot teach the returning of such an object to a CIM object manager.

Newkerk

Newkerk teaches a Common Agent that simply translates commands to be executed, and routes these commands to the appropriate destination, where the commands are used to direct the manipulation of an object. *Newkerk* thus teaches the manipulating of existing objects, but does not teach the creating of an object.

7

In the system of *Newkerk*, a Protocol Engine translates commands into the form supported by the Common Agent (*See* Fig. 1 and Protocol Engines, page 2). Such commands typically detail an operation to be carried out on an object, but are not objects themselves (page 1). These commands are then routed to the correct Managed Object Module, which then carries out these commands to manipulate the appropriate existing Object (*See* Fig. 1 and Managed Object Module, page 2). *Newkerk* therefore discloses a system that manipulates existing objects. *Newkerk* does not teach the creating of these objects, or returning these objects to a CIM object manager, however.

Ismael

Like *Newkerk*, *Ismael* teaches a network management system that translates commands and uses them to manipulate an existing object. Core management services 25-28 and m-beans 29, which are simply objects (Col. 6:18-25), each operate according to a specific protocol, such as HTML/HTTP, SNMP, etc. (*See, e.g.,* Col. 8:17-19). Commands are first sent to servers 32, 34, 36, 38, where they are translated to conform to that server's protocol (Col. 8:1-7). The server then operates on the appropriate object, via a framework 24, according to the translated command (Col. 6:39-47).

Like *Newkerk*, the system of *Ismael* simply translates commands and manipulates an existing object accordingly. The system of *Ismael* does not create objects, nor does it return them to a CIM object manager.

Accordingly, claims 1-17 should be considered patentable over *Nihart*, *Newkerk*, and *Ismael* for at least the reason that none of these references discloses the creation of objects, or the returning of such objects to a CIM object manager.

8

Applicant believes that all pending claims are allowable and respectfully requests a Notice of Allowance for this application from the Examiner. Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at (650) 314-5322.

Respectfully submitted,
BEYER WEAVER & THOMAS, LLP

Jon Y. Ikegami
Reg. No. 51,115

P.O. Box 778
Berkeley, CA 94704-0778

9